ARPANET Transition Opportunities and Gateway Considerations

Final Report

Jonathan B. Postel

Stephen D. Crocker

21 December 1977

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, California 90291

(213) 822-1511

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA

UNCLASSIFIED (18) SBIE (19) AD-E100 033

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER (12) 44p.
4. TITLE (and Subtitle) (6) ARPANET Transition Opportunities and Gateway Considerations.		5. ~~TYPE OF REPORT & PERIOD COVERED~~ (9) Final rept. 1 Sep 76 - 30 Jun 77 ~~6. PERFORMING ORG. REPORT NUMBER~~
7. AUTHOR(s) (10) Jonathan B. Postel Stephen D. Crocker		8. CONTRACT OR GRANT NUMBER(s) (15) DAHC15-72-C-0308, ARPA Order-2223
9. PERFORMING ORGANIZATION NAME AND ADDRESS USC/Information Sciences Institute 4676 Admiralty Way Marina del Rey, CA 90291		~~10. PROGRAM ELEMENT, PROJECT, TASK~~ AREA & WORK UNIT NUMBERS ID30431-VR
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communication Engineering Ctr. 1860 Wiehle Ave. Reston, VA 22090		(11) ~~12. REPORT DATE~~ 21 Dec~~ember~~ 1977 ~~13. NUMBER OF PAGES~~ 40
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document is approved for public release and sale, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Network, Gateway, Interconnection, ARPANET, AUTODIN II.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Issues in the transition of the ARPANET are discussed and a plan is outlined. The use of gateways is suggested, and issues related to them are discussed. The appendices include comparisons between alternate designs for three families of higher level protocols, host-to-host, terminal access, and file transfer.

407 952

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Contents

ARPANET Transition Opportunities and Gateway Considerations

I. Introduction

This report summarizes the the conclusions reached in the study task performed at ISI for the Defense Communication Agency (DCA) under the Integrated AUTODIN System Architecture (IASA) project.

The results can be divided into two principal groups: recommendations on the transition of the ARPANET and recommendations on the interconnection of networks via gateways.

The ARPANET is now supporting a large number of users performing (from the communications subnetwork point of view) routine work. While DCA now has the responsibility for operating the ARPANET, it is appropriate to ask if there are alternatives to its continued operation. Section II examines the possibilities and problems and presents a suggestion that interconnections between networks will be essential. Section III examines the issues involved with such interconnections, with the focus on the use of gateway devices.

Section IV summarizes the major findings; a set of appendices provide detailed informatinn on topics raised in the body of the report.

This report represents the views of the authors only and does not necessarily represent the view of ISI or the government. No inference should be drawn from this report concerning current or future policy of the government or the disposition of the ARPANET.

This report completes the study task performed at ISI as the Integrated AUTODIN System Architecture (IASA) project for the Defense Communication Agency (DCA) under contract DAHC15-72-C-0308 ARPA Order No. 2223.

II. Transition Opportunities

Introduction

This section suggests a plan for managing the ARPANET for the next several years and presents alternatives to it for providing intercomputer communication.

History

The ARPANET was developed by ARPA during 1968-1974 as a demonstration of computer resource sharing and packet-switching techniques for computer communication. From the initial four-node experimental network, the ARPANET has been expanded to roughly sixty nodes with roughly a hundred and fifty hosts [1] (an early plan envisioned the network completed at 19 nodes with one host per node). By every measure, the network has been a success and has fully demonstrated the efficacy of the original concept. As often happens with such successes, however, the experimental system is being exploited for many uses beyond its intended demonstration, and it is not possible to shut it down without incurring severe penalties. The ARPANET is currently used for a large variety of resource-sharing, management control, and cooperative research activities, which in turn have made it possible to carry out new research programs that would otherwise have been impossible. The National Software Works and the ACCAT Testbed are two such programs.

In 1975, because the ARPANET itself was no longer an experiment, ARPA turned control of the network over to DCA for operational management. Since that time, DCA has continued to operate the network and cause it to evolve slowly in response to known problems or needed capabilities. At the same time, the ARPANET had already served as a model for DCA's next major communication system, AUTODIN II. At present contracts for construction of AUTODIN II are in progress, and DCA now must contemplate what to do with the ARPANET when AUTODIN II becomes operational.

Discussion

Although the ARPANET served as a model for AUTODIN II, there are sharp differences between the two. These differences, which preclude simple plans such as including the ARPANET as a subpiece of AUTODIN II or moving all current ARPANET users onto AUTODIN II, are primarily the following:

1. AUTODIN II is required to handle classified traffic and to provide specified levels of service in times of stress. In contrast, the ARPANET is distributed among a large number of installations, including many universities ; it would be difficult to protect from attempts to disable the entire system.

2. Users of AUTODIN II are required to be military installations or installations engaged

in military projects approved on a case-by-case basis. Most of of the current ARPANET users would not be eligible.

This list can be extended, but the point is clear enough: neither the implementations nor the user groups of these systems can be melded together in any simple fashion. At the same time, the utility of the ARPANET is too high to dispense with it without replacement.

Why not operate the ARPANET indefinitely? While this seemingly obvious choice has the merit of preserving the environment for research programs in progress, it has several defects:

1. Because population of people and sites that need to communicate is growing, indefinite operation of the ARPANET implies its indefinite expansion. No resources exist within the government for this expansion.

2. As the network expands, the government will become increasingly vulnerable to criticism that it is competing with industry.

3. Many of the people and organizations that need to communicate with others on the network have their own funds to pay for the communication. At present, there is no legal mechanism for the government to collect funds from private users.

4. Because hardware used in the IMPs is nearly obsolete, maintenance will become increasingly difficult and expensive. More cost-effective hardware is readily available and, its use could lower the cost of operation of the network.

We are thus faced with a trilemma: The ARPANET is too useful to shut down, too expensive and limited to fund indefinitely and cannot--because of its implementation and user community--simply be absorbed by AUTODIN II.

Fortunately, there is a technical development that is providing other options: as new networks are being built, research is being carried out on how to interconnect different networks. Therefore, we now consider the future of the ARPANET in the context of possible interconnections to other networks.

The Plan

As AUTODIN II is built, some AUTODIN II users will need to talk with current ARPANET users; thus AUTODIN II and the ARPANET will need to be interconnected. Some of the parameters of this interconnection are explored below. Similarly, as commercial networks grow, current ARPANET users will need to talk with their users, and these networks will have to be interconnected.

What is the nature of the interconnection? If users are to communicate effectively, it must include compatible protocols for the basic activities: mail, file transfer and terminal access. If all of these activities can be made to work across networks, then most users will be able to use resources on either network without difficulty; the number of options for transition management of the ARPANET becomes quite large. However, it is possible that only mail transfer can be made to work efficiently in the near future, and thus the transition from the current state to any other state may be much harder.

Given a maximal set of options, what would we like to do with the ARPANET? The answer depends upon which set of users we look at, which today fall into two major categories. Most of the users use the network as an operational instrument; they can be served by any other network that provides the same services. A much smaller group is actively experimenting with the network and is expanding network technology; they require an experimental vehicle to continue their research.

Continuing with our assumption that all facilities are available across networks by internetting, the obvious strategy is to inhibit entry of new users and hosts onto the ARPANET, interconnect the network to one or more commercial networks and let the latter take over the nonmilitary users and sites. At the same time, AUTODIN II can connect to the ARPANET and internet with its sites. In cases involving security considerations, AUTODIN II would simply have to assume that the users on the ARPANET are uncleared.

Eventually, existing users could be moved from the ARPANET to AUTODIN II or commercial networks, and the ARPANET decreased in size. The rate of decrease and the limit on it depend on the need for the ARPANET as an experimental vehicle. When the need ceases altogether, the ARPANET can be shut down and scrapped.

As an aside, we can mention that scrapping the hardware of the ARPANET is the only viable option for getting rid it, since it will not be feasible to turn the equipment over to another operator or to reprogram the equipment in order to utilize it in another network. By the time the ARPANET is shut down, operational costs will be dominant. Replacing the IMPs will inexpensive in comparison to the operating costs. Moreover, there seems to be no viable alternative that does not involve overwhelming difficulties. For example, the sale of the Alaskan telecommunications facility to RCA required an act of Congress!

We now turn to the problem of transition of the ARPANET when only internet mail can be made to work efficiently. Basically, users would be moved to the networks that contain the hosts to which they need direct access. Interim operation requires that some hosts be attached to more than one network. As soon as a host is double-netted, users can start to move from the old to the new. When all have made the transition, the connection to the old network may be dropped; this will lead to a partitioning of users into quite separate groups corresponding to their primary network.

One countertrend to this idea is the possibility that everyone will need access to public networks. If so, AUTODIN II would be only a specialized network within the military hierarchy, like the military telephone system (AUTOVON) Almost everyone in the military with access to AUTOVON also has absolutely essential access to commercial telephone service. If data transfer becomes commonplace in everyday business life, then the same dynamics will prevail over AUTODIN II.

Exploration of internet mail facilities is currently planned under ARPA sponsorship and should provide a reasonable benchmark on the general internet problems. Coordination with ARPA is strongly suggested.

Comments on Alternatives

During the course of this project the Defense Communication Engineering Center (DCEC) prepared a list of possible strategies [2] to which the plan above and the comments below are generally responsive; in addition, point-by-point brief comments can be found in Appendix F.

Replacing the ARPANET with Other Networks

In a discussion of the future of the ARPANET, it is helpful to divide the current use of the ARPANET into two classes; experimental and operational.

A small number of users of the ARPANET utilize it as an experimental vehicle, including testing of new routing algorithms, interfacing with speech and packet radio experiments, etc. It is extremely unlikely that this type of use could be accommodated on any other system. For experimental use, either the ARPANET or some other private, experimental network will be necessary. It may be possible to use a network much smaller than the ARPANET, but it will *not* be possible to use a commercial network or AUTODIN II. Serious discussion of experimental use in the future requires ARPA's participation. As long as ARPA (or DCA or other agencies) have plans for experimenting with the details of the network, some experimental vehicle is necessary.

Operational use is another matter. Most of the current usage of the ARPANET is indeed operational--at least in the sense that users send data and expect it to get where it is supposed to. Most users are not interested in exact measurement of the throughput, delay or other statistics nor do they desire to perturb the implementation of the network to accomplish specific tasks.

While the majority of users on the ARPANET derive benefit from the ability to intercommunicate, there is no apparent reason that most of these users must use the ARPANET when equal or better facilities are available on other networks. It is only required that they be able to intercommunicate with present effectiveness. If other systems were available to these users, no apparent technical reason would prevent replacement of the ARPANET by one or more of these other systems.

It is quite reasonable to ask what it would take to move these users off the ARPANET. If it were just a matter of moving *all* of the users to another network, it would only be necessary to find (either by construction or procurement) another network with similar performance and cost. However, some of the current users would be attached to AUTODIN II, while others clearly would not; thus it is necessary to worry about interconnection as well as equivalent service levels.

Whether or not there is exists another system that provides service equivalent to the ARPANET remains an open question. There is no quantitative measure of the quaity of service ARPANET users now receive. (See Appendix E for more on equivalent service.)

Telephone use provides a clue on the interconnection problem. Only military sites (and related agencies) have access to AUTOVON. Everyone else uses the public telephone system. How do AUTOVON subscribers communicate with non-DoD people? Every AUTOVON subscriber is also a public telephone system subscriber! Thus, AUTOVON is a private network among the military, but it does not remove the need for everyone to subscribe to the same public network.

Is the same thing necessary for data traffic? Essentially, yes. The only modification possible is that the military can interconnect its own network to the public network and provide "local distribution." This is what is done within the military now for the voice network; AT&T provides service up to some boundary and then DCA or the Services take over from there. However, the end user can't tell where the boundary is. He dials area codes and the like as if he were connected to the regular public network. Callers from the outside also don't perceive the boundary.

To summarize the major point: military users of AUTODIN II will need access to public data networks. This access must be provided separately or it must be available as an option within the internal system.

To return to the question of what kind of service is acceptable to the users. As we remarked earlier, it is necessary to preserve the level of service now available on the ARPANET. (While improvement of the service is also desirable, the essential requirement is that the service not deteriorate.)

Exactly what is meant by equivalent service?

> The first criterion is that the protocols and functional capabilities must mate with current host software. Ideally, public networks would use the same protocols now in use on the ARPANET; failing that, the public network would offer an equivalent service which can be adapted to with only minor modification by the current hosts. The service offered by Telenet, Tymnet and others do not meet this criterion, but it is expected that they could extend their system to do so with only modest effort, if appropriately motivated.
>
> The second criterion is whether the performance and costs are acceptable. Slow file transfers, long echo delays or greatly increased costs are not acceptable.

The determination of these service parameters is discussed further in Section III.

The commercial networks differ from the ARPANET in several ways, a primary difference being that the ARPANET provides a datagram packet communication, while the commercial networks provide virtual call packet services. The ARPANET uses the BBN 1822 host to switch interface, while the commercial networks use the CCITT X.25 interface. Perhaps more serious is the lack of standardized higher level protocols, such as FTP, and the lack of generality in the terminal access procedures in the commercial networks. Another concern is the provision for the types of service needed to support projects like the National Software Works or the speech communication research effort.

Another factor in transferring ARPANET users to other systems is that we are not just transferring people but also a collection of host computers and software. It is essential that no significant changes be necessary to this body of software, including the host-to-host and higher level protocols.

A very considerable effort has been put into the development of the network-related software in the hosts. Significant costs would be associated with any retrofit of this software to the conventions and protocols of another network. The funding of such a conversion effort at the host sites would have to be carefully thought out and arranged with the host organizations before any movement of hosts from the ARPANET were put into effect.

We must take into account the economic and security aspects of the current environment. There is a strong economic motivation to combine the operation of the ARPANET with other systems, especially AUTODIN II. There may be security problems with this approach, and there may be important policy reasons for not allowing some legitimate classes of ARPANET users access to AUTODIN II.

If users are split between AUTODIN II and a commercial public data network intercommunication may be needed between those sets of people, which introduces the possibility that special gateways must be constructed to interface the two networks (see Section III).

Use of AUTODIN II for ARPANET Trunks

Another approach is the possibility of sharing resources between the ARPANET and AUTODIN II, yet operating them as separate networks in the user's view. There would not be interoperability due to incompatibilities in the higher level protocols. This type of resource sharing would clearly take design and development of hardware and software, and there may be difficult performance and security issues.

Using AUTODIN II for long-haul communication in place of the cross-country lines in the ARPANET is sure to cause some pain. The cross-country delay will be longer, and this may have serious impact on interactive applications, e.g., Telnet.

There are two ways to implement this idea, either with or without the knowledge of the IMPs. The interface between the current IMPs and AUTODIN II can take place at the modem interface level. The IMPs would send packets through AUTODIN II as if it were just another line and would have no knowledge of AUTODIN II. The other possibility is to modify the IMP code to explicitly route packets into AUTODIN II, in a fashion similar to the strategies used to connect the ARPANET to the satellite network.

The first approach is simpler and cleaner, but probably won't work. The IMPs know a lot about the delay, bandwidth and error characteristics of the lines connecting them, and it is unlikely that AUTODIN II can be made to look enough like a line to make this plan work. For example, if packets are not acknowledged from the neighboring IMP within a certain length of time, the packet is considered lost and a new copy is transmitted, often on a different path. Since the normal time delays are just transmission time, any connection through AUTODIN II is going to look like a very long line. The second approach is necessarily more expensive, since it involves modification of the IMP code and attendant testing and maintenance. If the second plan is seriously considered, BBN must certainly be consulted. In any case, the NCC operation will undergo a tremendous change, and one should plan for that too.

It is not clear that using AUTODIN II for trunking of ARPANET traffic would be permitted under the security and precedence policies. At best, all traffic to and from the ARPANET would have to be treated by AUTODIN II as being of the most unsecure and least important (low precedence) type.

Using AUTODIN II for cross-country communication is only one part of a much bigger problem. Eventually, AUTODIN II, the ARPANET, and one or more commercial services will have to be interconnected so that users of one system can communicate with users of the other systems. There are important protocol and implementation problems to be solved, to say nothing of the administrative issues. (See Section III.)

As data communication becomes more and more commonplace, commercial service will become more important. All users need access to some commercial service, but only selected subsets need access to either AUTODIN II or the ARPANET.

Accordingly, it is important to begin interaction with the commercial services as soon as possible. Many of the current ARPANET users might reasonably be connected to commercial data networks; a few of these might also be connected to the ARPANET for experimental purposes. Use of AUTODIN II should be restricted to just those needing the security or emergency operation of a military network, but all of these users should also have access to regular commercial services.

III. Gateway Considerations

Several large scale-computer networks now exist, and more are being built each year. In many cases, users connected to one of these networks will need to interact -- or have their computers interact -- with computers on other networks. There is no possibility that one single network will immediately supplant the many networks now in use or under construction, so users will have to contend with a variety of networks for at least several years. Techniques are needed for effective interconnection of networks and transmission of data among host computers attached to them.

The problem of connecting networks is similar to the problem of connecting computers, (i.e., designing a network). The basic problem in network design has two essential components: what facilities should be built into the communication subnet, and what strategies should the hosts use to make effective use its facilities. Similarly, the internetwork design problem has two components: how should gateways be built to provide transmission of data from one network to another without compromising the security and reliability of the participating networks and maximizing the throughput or other performance measures, and what strategies should the hosts use for communicating with other hosts in distant networks? Three specific types of network interconnection called "hidden gateways", "high level gateways", and "two connected hosts" are discussed briefly in Appendix G.

A network or internetwork system provides a set of transmission facilities or services; the host systems adopt strategies for utilizing those services to communicate among themselves.

In the near future, several interconnections are planned by ARPA among existing networks or networks under construction. Central to all of these plans is the ARPANET, which is of interest for two reasons. First, it is designed as an experimental network and thus is an obvious candidate for early experiments in interconnecting networks. Second, it is serving a large number of users who will need equivalent service if the ARPANET is ever phased out. With the development of AUTODIN II to serve military users and commercial networks to serve others, the role of the ARPANET *as a service* is indeed subject to review.

The interconnection of the ARPANET or AUTODIN II and a commercial network brings up the question of interface and protocol compatibility. The commercial networks are moving to the use of the CCITT X.25 interface and to CCITT protocols quite different from those used in the ARPANET or AUTODIN II.

In order to discuss if users can be moved from the ARPANET to other networks, one needs to know how users currently use the ARPANET. Usage falls into a few major categories: remote terminal access, remote job access (batch), mail transmissions, file transmissions, communication of status, and process-to-process communication. One needs to have metrics in each of these categories for performance as perceived by users.

Three areas of concern have been identified. First, the design of gateways to interconnect existing networks and related extensions to the host-to-host protocols. Second, the examination of strategies available to hosts for use of higher protocol level, function-oriented applications in internet environments. Third, the specification of metrics of network usage at typical hosts and the characterization of that usage for determining equivalence of service and gateway performance.

We have identified three strategies that apply to both of the first two areas. One can bring independently developed systems into an interworking environment by one of three techniques: identifying *subsets* of services or capabilities that interwork without change; constructing mechanisms that *transform* the interactions such that each system produces and receives its outputs and inputs exactly as it expects; and modifying the systems such that as they evolve, they *converge* toward a common standard.

Gateway and Host Protocols

The first concern is the design of gateways to interconnect existing networks, and extensions to the host-to-host protocols. This requires that flow-control and routing schemes be developed to connect the two networks and that performance measures be chosen to evaluate the effectiveness of the interconnection.

Appendix A presents a comparison of TCP [3] with NCP [4] showing the difficulty of direct communication between hosts that use these two host-level protocols.

It is important to determine the minimum functions required of a gateway for each type of service desired. The essential point seems to be that care must be taken not to overly restrict the possible uses of a set of interconnected networks by a too encompassing gateway design.

The gateway may evolve into a mechanism with knowledge of various protocol levels, i.e., as a more specific type of service is requested more levels of protocol are invoked, with the protocol at each level specific to the type of service called for.

Another important area is the provision for addressing; a mechanism that provides for an extensible address or source specified routing will apparently be necessary.

Approaching the problem of host-to-host communication through an internetwork gateway system via the three strategies cited above (subset, transform, converge) produces the following recommendations:

Subset:

It is unlikely that an adequate interworking system could be achieved at the host level by the subset strategy (see Appendix A).

Transform:

Given the current state of network development, perhaps the most practical way of proceeding in the short term is the transform strategy. However, to say that this is perhaps the most practical is not to say that it is easy. The comparison in Appendix A should make the difficulties clear. Experiments are necessary to confirm the requirements for gateway design; they should be designed to include the transformation from one host-level protocol to another, e.g., ARPANET NCP to AUTODIN II TCP.

Converge:

In the long term the most effective strategy is the convergence of the various systems to a common standard, but such a standard must be based on an accurate understanding of the requirements for communications protocols since it will be very hard to change once it is achieved. Appendix D discusses some issues pertinent to such a protocol.

Host and Application Interoperability

The second area is the consideration of strategies to be used by the hosts independent of the network carrier implementations. The higher level protocols, e.g., virtual terminal and file transfer, are replicated on most of the various networks, but are often incompatible in detail. An examination of these two major protocol families shows the difficulties of internetwork communication at the higher levels. In particular, the Telnet protocol in the ARPANET [5] and the Scroll Mode Virtual Terminal protocol (SMVT) proposed for European Informatics Network (EIN) [6] are compared in Appendix B, and the FTP protocol in the ARPANET [7] and BTF proposed for EIN [8] are compared in Appendix C.

The problem of process-to-process communication through an internetwork system is also approached through the subset, transform, and converge strategies cited above.

Subset:

It seems that a marginal interworking system could be achieved at the process level by the subset strategy, particularly for the most basic types of terminal-to-host interaction, involving the simplest application programs. This approach does not appear to be worth pursuing, however, since it is inadequate for many application programs, sophisticated terminals, and higher level protocols.

Transform:

In each network a series of application level protocols are developed. For example, in the ARPANET there is Telnet, file transfer, and remote job entry. In AUTODIN II there is THP, and eventually a file transfer protocol. The parallel function of higher level protocols (e.g., the Telnet, THP correspondence), especially file transfer protocol, suggests that the transform method of providing an interworking system is quite practical. Inspection of both the virtual terminal and file transfer protocols shows many functions where fairly simple on-the-fly transforms can be provided; however, there are also a few functions that will require rather complex transforms--if transforms for those functions can be provided at all.

Converge:

In the higher level protocols it is relatively easy to operate two or more protocol modules that offer functionally equivalent services in parallel. The design of an ideal protocol with the goal its of eventual implementation in each of the interconnected networks in parallel with the locally developed protocols seems a reasonable long term goal. (This would have to be a very long term plan, coupled with action in the appropriate national and international standards organizations.)

Service Equivalence and Gateway Performance

Because, unfortunately, there is no clear picture of the performance of the present system, *as seen at the user level*, an immediate task is to find out how much traffic is generated by various uses of the network and what delay, throughput, cost and other relevant parameters characterize these uses. These measurements will provide the quantitative refinement to the qualitative requirements derived from examining the protocols.

The problem of determining what service the ARPANET users do receive is complicated because appropriate metrics and their values are difficult to identify. The traditional communication measures of delay and throughput may not accurately approximate the user's perception of service quality. For example, some users may be more sensitive to command completion time and consistency, some to smoothness of service. Perhaps a measure of acceptability based on factors of reliability, availability, responsiveness, and cost would be more realistic.

The development of such measures must take into account the problems of the random and uncontrollable load on the network. Measurements made under controlled conditions are preferable to passive observations.

Measures of service, and user-perceived values of those measures, should be determined, and the resulting information used in considering what service another should provide in order to be considered equivalent to the ARPANET. Some further comments on determining service equivalence appear in Appendix E.

An experimental gateway should be constructed and tested in the roles of (1) routing uninterpreted messages between two networks only, (2) performing host level protocol transformations, and (3) performing higher level protocol transformations. The performance of this gateway should be carefully measured. Two types of measurements are needed:

measures local to the gateway and measures of the impact on users perceptions of end-to-end service.

IV. Summary

Transition Opportunities

A practical transition plan for the ARPANET must provide for continued service to the numerous people using the ARPANET as an operational tool. Any substitution of service must provide the same ease of intercommunication and resource sharing without deterioration in the quality of the service. The suggested plan calls for the transfer of current ARPANET users to other networks while providing for their continued interaction via the interconnection of networks. In brief, the following steps are identified.

1. Identify one or more commercial networks that can supply service equivalent to the ARPANET.

2. Interconnect the ARPANET and the commercial network(s).

3. Interconnect the ARPANET and AUTODIN II.

4. Move the military users to either the commercial network(s) or to AUTODIN II.

5. Move the nonmilitary users to the commercial network(s).

6. When the alternate networks are providing equivalent or better services, reduce the size of the ARPANET to that needed by the small set of users performing network experiments.

The movement of users from one network to another is likely to also require the movement of hosts, which is likely to incur significant software costs. The alternative to abruptly moving a host is to connect it to both networks.

Gateway Considerations

The design of gateways between networks brings up many issues. A particularly important one in consideration of achieving a near-term network interoperability is the gateway's ability to provide protocol transformations. In the near term the possibility for users of one network to share the resources of a host on another network will depend on the gateway's ability to map or transform the underlying protocols used in the two networks into each other.

References

[1] Feinler, E., ARPANET Resource Handbook, NIC 23200, Stanford Research Institute, Menlo Park, California, September 1975. And Feinler, E., "Hosts and IMPs," ARPANET Message, SRI International, 12 December 1977.

[2] Hawrylko W., "Strategies Concerning the Eventual Disposition of the ARPANET," ARPANET Message, Defense Communication Engineering Center, Reston, Virginia, 17 February 1975.

[3] Cerf, V. Y. Dalal, and C. Sunshine. "Specification of an Internet Transmission Control Program," INWG General 72, RFC 675, Revised December 1974.

[4] McKenzie, A. "Host/Host Protocol for the ARPANET," ARPANET Protocol Handbook, Network Information Center, NIC 7104, Revised April 1976.

[5] McKenzie, A. "Telnet Protocol Specification," ARPANET Protocol Handbook, Network Information Center, NIC 7104, Revised April 1976.

[6] Schicker, P. and H. Zimmermann. "Proposal for a Scroll Mode Virtual Terminal," Centre Coordination Group, European Informatics Networks, EIN/CCG/77-02, INWG Protocol 62, January 1977.

[7] Neigus, N. "File Transfer Protocol for the ARPANET," ARPANET Protocol Handbook, Network Information Center, NIC 7104, Revised April 1976.

[8] Schicker, P., A. Duenki, and W. Baechi. "Bulk Transfer Function (Proposal)," European Informatics Networks, EIN/ZHR/75/20, INWG Protocol 31, January 1977.

[9] Postel, J., L. Garlick, and R. Rom, "Transmission Control Protocol Specification," Report No. 35938, Defense Communication Engineering Center, 15 July 1977.

[10] Postel, J., L. Garlick, and R. Rom, "Terminal-to-Host Protocol Specification," Report No. 35940, Defense Communication Engineering Center, 15 July 1977.

[11] CCITT, "International Alphabet No. 5," CCITT White Book, Volume VIII, Recommendation V3, Mar del Plata, 1968.

[12] ANSI, "USA Standard Code for Information Interchange," USAS X3.4-1968, American National Standards Institute, 1430 Broadway, New York, New York, 10018.

Appendix A

Background on TCP and NCP

Some of the main points of two host-level protocols are described here to indicate some of the issues involved in gateway design. The two protocols are compared along a set of attributes common to all host-level protocols. The particular protocols compared are the TCP [3] and the NCP [4], both used in the ARPANET. Because the TCP used in AUTODIN II [9] is quite similar to that in [3], this comparison holds for the consideration of the interconnection of ARPANET and AUTODIN II. The evolving international standards are likely to be derivative of TCP-like European protocols.

Control Machinery

TCP

Control information is communicated only in the headers of data messages, most via flag bits in each header.

NCP

Control information is exchanged between NCPs via control messages on a logical channel distinct from those used for data communication. The control information is primarily carried as parameter fields in what would otherwise be the data portion of the message. Most control messages are related to particular data channels via the link parameter.

Addressing

TCP

Sockets, defined to be the addressable entities, are the end points of logical connections. Each host provides a local mapping between sockets and the input and output ports of processes. TCP connections may be full duplex, and one socket may participate in many connections. A connection is uniquely identified only by the pair of sockets at its end points.

NCP

The NCP defines sockets as the addressing points in the process-to-process communication system it implements. Each host provides a local mapping between sockets and process input and output ports. Sockets are the end points of simplex connections, that is, data may flow only one direction on a connection, and a connection may be uniquely identified by a single socket. This means that a socket can participate in only one connection at a time.

Error Control

TCP

A checksum is placed on each message sent by the sending TCP and verified by the receiving TCP.

NCP

The NCP was designed in the context of the ARPANET, which is a very reliable communication system. The NCP does not provide for error control, but relies on the ARPANET to deliver messages correctly and in order, without duplication or omission.

Flow Control

TCP

TCP uses a window mechanism. In each data message the header contains an indication of the amount of data the sender of this message is willing to receive in future messages travelling the opposite direction. The window concept comes from the expression of the indication as a portion of the sequence number space. TCP uses sequence numbers associated with each data octet, and the window information is communicated as the highest numbered octet currently acceptable to the receiver. The window is that part of the sequence number space between the last sequence number used by the sender and the highest sequence number acceptable to the receiver. As data is sent, the window becomes smaller, and as data is received and processed the data receiver increases the value of the highest acceptable sequence number and this higher value is carried in the window field of messages traveling in the other direction, whereupon the window becomes larger.

NCP

The NCP uses an allocation mechanism. The receiver of data messages on a per-connection basis advises the sender of data via control messages of the amount of data the the receiver will accept. The mechanism uses two counts: one of the number of bits of data and the other of the number of messages the receiver allows the sender to send. When the sender sends data each count is reduced by the number of data bits and the number of messages sent. The sender must never reduce either count below zero. The receiver, as it processes data, may send control messages containing increments to the counts, allowing more data to be sent.

The NCP also requires that on each logical channel (or connection) only one message may be in progress in the ARPANET. (In ARPANET terms the NCP must wait for a RFNM from one message before sending another message on the same link.)

Sequence Control

TCP

The sequence numbers used by TCP allow the receiver of messages to order them in the same order as sent by the sender. The sequence numbers also allow the detection of duplicate messages and the detection of missing messages.

NCP

The NCP does not provide a mechanism for ordering messages. Instead it relies on the communication subnet, the ARPANET, to deliver messages in order. The ARPANET does deliver messages in order per host-host pair.

Format

TCP

The only format constraints placed on user data is that transmitted data is carried in octets and that the data be blocked into letters, although there is no limit on the length of a letter. There is a way for the sending user to flag the end of a letter, and when this is done the receiving user will be notified of the end of letter flag as it reads the data. Each message carries a TCP header of 288 bits. In the ARPANET a message is limited to 8095 bits, including the 32 bit leader, so the user data portion would be 8095-288-32=7975 bits. In addition, TCP transmits data in octets, so the message must be a multiple of 8 bits, which reduces the maximum user data messages to 7968 bits (996 octets).

NCP

No format is imposed on user data transmitted via the NCP. In addition, most NCPs are implemented in such a way that the user is prevented from knowing where message boundaries occur. In the ARPANET messages are limited to 8095 bits. The NCP header on each message is 40 bits, and the ARPANET leader is 32 bits. The data space available to the user is then 8095-40-32=8023 bits. The NCP imposes no character set or byte size on the user, the byte size is limited to the range 1 through 255 bits per byte, and must be constant for the lifetime of a connection.

Throughput & Delay Effects

TCP

The window flow control strategy has potentially powerful effects on the throughput and delay of messages sent via TCP. A rapid variation of the window size could have undesirable effects on throughput. If the window size were to become zero even temporarily, delay would be substantial. The optimal window control policy is still a research topic.

NCP

The restriction of one outstanding message at a time per connection limits throughput by introducing a round trip delay between each message of the communication.

The allocation mechanism introduces the possibility of delays if either allocation count falls to zero. This could easily happen if the destination host has very small buffers or uses a buffering policy that limits it to processing only one or two messages at a time however little data the messages may contain.

Delays may be introduced because allocation control messages are concentrated on one logical channel when there are many separate logical data connections between a pair of hosts.

Summary

TCP

TCP was designed to operate robustly in the face of unreliable communication systems. All data and control information related to one conversation share the same logical channel. Connections end in sockets and are full duplex. TCP provides end to end error control using checksums. TCP's flow control is provided via a window mechanism based on sequence numbering of data octets, which is also used to order the data. Data is sent in octets in variable length letters. The window adjustment strategy may have significant effects on throughput and delay.

NCP

The NCP was designed to operate in the very reliable communication environment provided by the ARPANET. Data and control information for a connection travel on distinct logical channels. All control information for one host-host pair share the same logical channel. Connections end in sockets and are simplex. The NCP relies on the ARPANET for error control. Flow control is based on an allocation mechanism where positive counts of messages and data bits are incremented by the data receiver and decremented by the data sender. The NCP relies on the ARPANET for ordered delivery. Data may be sent in any of a wide range of byte sizes, message boundaries are not reported to the user. The allocation policy may have severe effects on the throughput and delay; throughput is limited by the requirement to wait for confirmation of delivery by the subnet.

Appendix B

Background on Virtual Terminal Protocols

Some of the main points of two virtual terminal protocols are described here to provide some indication of the issues involved in application level protocol conversion. The two protocols are compared along a set of attributes common to all virtual terminal protocols. The particular protocols compared are the ARPANET Telnet [5] and the proposed SMVT for EIN [6]. The THP used in AUTODIN II [10] is similar to Telnet, while the evolving international standards are likely to be influenced more strongly by SMVT.

Control Machinery

SMVT

Communication takes place in blocks, where each block is a sequence of octets. Each block begins with a block code that indicates the type of information in the block to be one of text, control, or parameter. Following the block code are a series of items, which may be variable length. Each item starts with an item code octet, followed by either a single octet value or an octet specifying the length and length octets of value. Interrupts are also used in the control mechanism. In the assumed underlying host-to-host protocol it is possible to send interrupts with 16 bits of associated data. These interrupt data bits are divided into two octets, the first being the interrupt code, and the second being the interrupt parameter.

The virtual terminal protocol operates in one of two modes: alternate mode, in which the right to send data is passed from one side to the other; and free running mode, in which either side may send data at any time.

There is a set of parameters that may be negotiated at the initiation of the connection only. The following is the negotiation scenario:

```
serving                                       using
host                                           host
-------                                       -----
<request parameter range> --->
                    <--- <indicate parameter range>
<select parameter values> --->
          <--- <agree/disagree on parameter values>
```

Telnet

Telnet treats the communication as a continuous stream of data characters with embedded control information flagged by an escape character. All control information is communicated in the data stream as a sequence of 8-bit bytes. There is also a resynchronization mechanism, which is discussed under Error Control.

There are some control signals that must be interpreted by all Telnet implementations. These are transmitted as single byte codes (preceded by the flag byte).

Telnet initially operates in an alternating mode with control passed by means of a "go ahead" control signal. An optional free running mode is available.

There is a set of optional features that are negotiated into use by a exchange of control messages. For example, let us use the option for controlling which side of the connection provides echoing of the the transmitted characters across the connection. The default is that no echoing is done across the connection and the echoing of the user's typed characters is provided by the terminal or the local host. If the situation called for the remote host to provide the echoing, the following negotiation might take place. We assume that the remote host recognizes the need for it to provide the echoing and that it initiates the negotiation.

```
local                        remote
host                           host
-----                        ------
          <--- <flag><will><echo>
<flag><do><echo> --->
```

Now the remote host will provide the echoing; presumably the local host has turned off the local echoing to prevent each character from being echoed twice.

If the user thought it inappropriate or the local host could not engage in this option, the dialog would be as follows:

```
local                        remote
host                           host
-----                        ------
          <--- <flag><will><echo>
<flag><dont><echo> --->
```

In this case the option is not used, and the situation continues as before the negotiation.

In general options negotiations are based on four control terms: <will>, <wont>, <do>, <dont>, with the following general interpretations:

<will> the sender of this command requests or confirms that it will perform the option.

<wont> the sender of this command refuses to perform the option or requests to discontinue performing the option.

<do> the sender of this command requests or confirms that the receiver of this command perform the option.

<dont> the sender of this command refuses to allow the receiver of this command to perform the option or requests the receiver of this command to discontinue performing the option.

In addition, some options have within them provision for subnegotiation.

Applications or User Interface

SMVT

The SMVT specification leaves open many aspects of the user interface. It is specified, though, that provision must be made for the user to enter all 95 International Alphabet Number 5 [11] graphic codes as data to be transmitted, and that the user have a way of specifying certain control signals.

An implementation must of course provide a means for the user to specify the address (host and socket) to connect to, or which parameters should be negotiated, or which local terminal modes should be used, but the command interface for these is for each host to determine.

Telnet

The Telnet specification leaves open many aspects of the user interface. It is specified that provision must be made for the user to enter all the ASCII [12] codes as data to be transmitted, and that the user have a way of specifying certain Telnet control signals.

An implementation must of course provide a means for the user to specify the address (host and socket) to connect to, or which Telnet options should be negotiated, or which local terminal modes should be used, but the command interface for these is for each host to determine.

Data Representation

SMVT

SMVT specifies that the data shall be 7 bit IA5 characters right justified in octets with the unused bit set to zero. Data is transmitted in a segments (one kind of text item), with a maximum length of 127 octets. Text items may be grouped blocks of indefinite length for transmission. Blocks may be constructed and transmitted without regard for letter boundaries from the SMVT's point of view.

Telnet

Telnet specifies that the data shall be 7 bit ASCII characters right justified in 8 bit bytes with the unused bit set to zero. Other character sets may be used under the negotiated options. Data is transmitted in a undivided stream, with no blocks or records. There is no limit on the length of the data stream. Telnet does not expect to use message boundary information, since the ARPANET host-to-host protocol (NCP) suppresses it.

Note that IA5 and ASCII are essentially identical.

Error Control

SMVT

SMVT relies on the host-to-host protocol to provide error-free communications of ordered data. SMVT does provide a resynchronization mechanism called CLEAR that involves uses of the interrupt facility provided by host-to-host protocol together with a special item in the data stream.

Telnet

Telnet relies on the host-to-host protocol to provide error-free communications of ordered data. Telnet views the communication as a continuous stream of characters. Telnet does provide a user controlled resynchronization mechanism called SYNCH that involves uses of the interrupt facility provided by host-to-host protocol together with a special mark in the data stream.

Delay & Throughput Effects

SMVT

SMVT does not of necessity introduce any delay or throughput effects. The transmission delays in the network will have an effect on the user. In the case of a user locally connected to the serving system and employing the alternate mode of interaction, when the user types the final character the serving computer is immediately signalled to take action. In the network case when the user types the final character the data begins transmission through the network. Thus the user sees the response time to commands lengthened by the network transmission time.

Telnet

Like SMVT, Telnet does not of necessity introduce any delay or throughput effects. However, users of an interactive computer system will notice a difference between responsiveness when directly connected to the serving computer and when accessing the same serving computer via Telnet and a communications network.

Summary

SMVT

SMVT was designed in parallel with virtual terminal protocols for more powerful data entry display terminals and the command codes and formats are compatible with those more elaborate protocols. A major concept in the design of this set of virtual terminal protocols is the idea that the application program writes into a data structure in the virtual terminal, which then portrays changes in the data structure to the user via the real terminal. Data and control exchanges are formatted into blocks of items. The data exchange mode may be either alternate or free running, the choice being made at the beginning of the session. Other parameters may be also negotiated at the beginning of the session. Use of IA5 is required. SMVT provides a CLEAR procedure for resynchronizing the exchange of information.

Telnet

Telnet was designed with teletype-like terminals, with no internal memory, as the prototypical terminal. Information flows between the protocol modules as continuous character streams with control information flagged by special escape characters. Options may be negotiated at any time. The flow of data is initially alternate but may be negotiated to free running. The ASCII character set is required, though others might be negotiated. A SYNCH mechanism is provided to resynchronize the processing of data flowing in one direction or the other.

Appendix C

Background on File Transfer Protocols

Some of the main points of two file transfer protocols are described here to indicate some of the issues involved in application level protocol conversion. The two protocols are compared along a set of attributes common to all file transfer protocols. The particular protocols compared are the ARPANET FTP [7] and the proposed BTF for EIN [8]. The file transfer protocol for use in AUTODIN II has not been specified as yet, and there has been no noticeable activity on file transfer in the standards organizations.

Control Machinery

BTF

Control information is exchanged via "control directives" which are strings of octets, each string being composed of an operation code followed by zero or more arguments. The BTF programs are equals, and either may send a request or command to the other. Each request has one reply. Requests and replies are in the same format. Control and data may be sent on the same logical channel or on two distinct logical channels.

FTP

In FTP one program is the user and the other the server; control requests or commands are sent from the former to the latter in the form of a character string in which the first word is the command word and subsequent characters are arguments. Each request is limited to one line. The server replies to requests in the form of a character string of which the first three characters are a reply code and the remainder is a message to be presented to a human user if necessary. The reply code is designed to allow a program to determine if the preceding request was successful or not. This request and reply control mechanism was designed to be used directly by human users as well as to be interpreted by programs. In FTP control information travels on a logical channel distinct from that used by the data.

Applications or User Interface

BTF

The BTF specification suggests that the user or application program call on the BTF via the function calls described below. It up to each host to provide this or an equivalent mechanism.

The suggested calls are:

OPEN <device reference> <access mode> <device position>

CLOSE

SEND <device reference> <amount>

RECEIVE <device reference> <amount>

COUPLE <device reference> <device reference>

where

<device reference> identifies the host, file or device, and indicates access parameters;

<access mode> indicates which of read, write, alter, and extend access is desired;

<device position> indicates which of rewind, current, append, and resume positioning is desired;

<amount> indicates the number of octets to be transmitted.

- FTP

Because the FTP-to-FTP control requests and replies were designed to be used directly by human users as well as programs, the applications or user interface is very similar to the control requests and replies in most hosts.

Some typical requests and replies are:

```
USER jones
331 user name okay, need password
PASS secret
230 user logged on, proceed
RETR <jones>program.source-code;3
150 file status okay, about to open data connection
226 closing data connection, requested file action successful
BYE
221 user logged off
```

Addressing

BTF

In BTF the addressable entity is specified by a device reference, which includes information to specify the host, the device or file, and information that allows a BTF to determine if this access is allowed by the administrative access controls and if it is sensible. (For example, an attempt to write to a card reader is not sensible.) The BTF appears to be designed to transfer whole files, except that a transfer may be appended to an existing file.

FTP

In FTP the addressable entity is specified by a pathname, which is defined to be a character string. In practice this is a filename, and FTP deals in whole files. There is a provision for appending a transmission to an existing file.

Access and Authorization

BTF

Access control and authorization checking are embedded in the device reference information communicated via the user interface OPEN call and the OP control directive.

FTP

Access control is provided by having the user identified via a set of requests (USER, PASS, ACCT) and allowing each FTP program to determine if administrative access is allowed by this user to the files specified in the file action commands.

Data Representation

BTF

The strategy in BTF is to define a set of atomic data elements and allow the transmission strings to be built up of them. The basic elements are the following:

Character
Bit String
Integer
Real
Decimal

Data is transmitted in zones which are up to 128 octets of element directives and data.

The is no limit on the number of zones in a transmission. There is a directive to indicate the end of data.

FTP

The strategy of FTP is to describe the form of the file to be transmitted along several quasi-independent dimensions. The dimensions used by FTP are described in the following, together with the possible values.

Byte Size
  1 through 255
Type
  (ASCII or EBCDIC) and (non-print or telnet or ASA)
  Image
  Local Byte (of size) N
Structure
  File
  Record
Mode
  Stream
  Block
  Compressed

There is no limit on the length of a transmission. In stream mode there is no length indication at all; the end of transmission is indicated by closing the logical channel. In block or compressed modes there are descriptors that occur at least every 2**16-1 bytes (the bytes may be up to 255 bits each), but there is no limit on the number of blocks. There is a descriptor to indicate end of transmission.

Error Control

BTF

BTF relies on the underlying host-to-host protocol to provide error-free communication, though it does provide for restarting a transmission at an intermediate point if that is desirable. For example, if a paper jam occured when transmitting to a line printer, it would be desirable to restart the transmission (after the jam was fixed) at the point where the listing became unusable. The restart mechanism works as follows:

> The data sender inserts markers in the data stream, which are recorded by the data receiver. If a restart is necessary, the data receiver then tells the data sender the marker at which to start. A marker must be placed at least as often as every 4096 octets.

FTP

FTP relies on the underlying host-to-host protocol to provide error-free communication. In block or compressed mode, FTP provides a restart mechanism as follows:

> The data sender inserts markers in the data stream, which are recorded by the data receiver. If a restart is necessary, the data receiver then tells the data sender the marker at which to start. There is no requirement for the frequency at which markers are to be placed. There is provision for communicating the markers to the human user.

Throughput & Delay Effects

BTF

This data transfer mechanism should have no detrimental effects on throughput or delay. There might be a potential problem of translating data on the fly from internal format to transmission format and vice versa.

FTP

The FTP data transmission mechanism should have no detrimental effects on throughput or delay. It is possible, for some combination of modes, that the conversion from internal form to transmission format and vice versa might be a problem. . In some cases the control mechanism introduces unnecessary delays by using several interactions where one would be sufficient; this deficiency cannot be corrected without significant changes to the request and reply format.

Summary

BTF

BTF modules are equals that exchange request and replies in the form of control directives. Control and data may share one logical channel or use distinct logical channels. The user or application interface is via a set of system calls. Whole files are accessed, but devices can also be addressed. Access control information is contained in the same control directive that addresses a file. The data representation is based on a set of atomic data elements. A transmission restart procedure may always be used.

FTP

FTP modules are distinctly user and server modules. Requests from the former and replies from the latter are quite different in format. The control (request and replies) travel on a distinct logical channel from the data. The user or applications interface is essentially the language of the request and replies which are character strings. Whole files are addressed. Access controls are transmitted in user identifying requests distinct from the file addressing requests. The data representation is based on a selection of attributes from several quasi-independent sets. A transmission restart mechanism may be optionally employed with some transmission modes.

Appendix D

Protocol Model

The following model of a communication protocol is presented with the aim of clarifying the essential nature of the exchange of control information as compared to the exchange of data information.

Most protocols are designed to carry data between processes. Protocol modules are implemented to ease the burden on the process and to ensure reliable transmission of the data. Protocol modules generally implement mechanisms for error, flow, and sequence control for the data traffic. From time to time one process or another is temporarily unable to accept data. This is generally reflected in the protocol module in the flow control machinery for the data transmission. This is as it should be; however, in some protocols a problem arises because the exchange of control information between the protocol modules themselves is coupled to the flow of data between the processes.

A protocol must not block the flow of control information between protocol modules at any time for any reason, which implies that the protocol should be designed first to provide reliable non-blocking communication of control information between protocol modules, and secondarily to carry data between processes. The design of a protocol must not allow the mechanisms for reliable communication of control to be corrupted by the data transmission provisions. Also for two protocol modules to be in communication they must exchange messages at some minimum time interval.

An example of such a protocol is developed below.

> To provide for reliable communication each message carries a checksum verified by the receiver. To be sure that messages are processed in order, each message carries a sequence number. To be sure that each message sent has been received without error, each correctly received message is acknowledged by its sequence number. If the sender of a message does not receive an acknowledgment for that message within a reasonable time, it retransmits that message.
>
> The following simple system is a protocol that exchanges control information only. Suppose that each of two protocol modules has two cells called my-cell and your-cell. The value of your-cell is set by messages received from the other protocol module. The value of my-cell is set by a computation that might be based on the past values of my-cell, the value of your-cell, or a clock. Each time that the value of my-cell is changed a message is sent to the other protocol module.
>
> In this system messages are exchanged such that each protocol module has an up-to-date record of the cell the other module controls.

We add to this system the sending of process-to-process data by attaching the data to the control messages. We simply add fields to the message to contain the data. The field could be empty if there is no data to send at a time when a control message is to be sent. At times there may be data to send when there is no reason to send a control message. One might choose to force the sending of a control message to carry the data. This approach is rejected because it corrupts the control machinery.

Instead the new data is sent in a message with its control part an exact copy of the previously sent message. To reliably transmit the data using this method we introduce a two-part sequence number, which is composed of a control part (as before) and a data part, which sequences the data relative to the control sequence number. Acknowledgments must be sent for each correctly received message and now must carry both parts of the sequence number.

Appendix E

Service Equivalence

1. The immediate concern is whether or not there is a commercial alternative to the ARPANET to which existing subscribers could switch.

2. The availability of an alternative must be determined on the basis of some measure of service equivalency between the ARPANET and the proposed alternative.

3. Equivalency should not be evaluated on the basis of total duplication of service capabilities, but on some measure of overall acceptability.

4. Acceptability measures must include certain pragmatic variables such as cost, viability, reliability and consequences. If the ARPANET is to survive as a much diminished, experimental-only network, what will be the impact on projects truly experimenting with networking technology?

5. Network measurement requires careful control of host responsiveness and user behavior, and--to whatever degree possible--control of the common carriers serving the network as well. Most network measurements taken in the past have been uncontrolled observations, with no way of partitioning and thus assessing the variance in performance associated with those variables that determine performance.

6. Whatever measures are determined to be valid characterizations of what good service means to a user, data should be collected within the framework of a valid scientific experiment, not as mere observation.

Appendix F

Strategies Concerning the Eventual Disposition of the ARPANET

The following seven strategies were identified by DCEC [2]. While the main body of this report speaks to these strategies and indicates their advantages and drawbacks, in this appendix brief comments are made on each point.

1. Leave the ARPANET as presently constituted and simply gateway it to AUTODIN II. DoD users could then choose to remain on the ARPANET or become AUTODIN II subscribers. DCA would still be required to provide operational and transmission facility management of the ARPANET.

> This is a quite desirable strategy from the point of view of the users of the ARPANET. It has two drawbacks: cost and growth. The cost of operating the ARPANET is likely to grow even if it is stable in size (measured in users, hosts, or packets transmitted), since as the hardware ages it will become less reliable. The ARPANET is likely to grow. There will be pressure to add users because they need to interact with and share resources with users already on the ARPANET.

2. Partition and regionalize the ARPANET and provide long haul trunking via gateways to *AUTODIN II.* DCA *would still* have to manage some separate ARPANET transmission facilities and operationally manage the network.

> If one could do this invisibly to the ARPANET user it would be clever way to reduce the cost of operation, provided that the cost of such trunking is less via AUTODIN II than the current arrangement. There are serious questions about the possibility of such an invisible substitution of long haul facilities. This strategy is specifically addressed in Section II. A major concern is the achievement of end-to-end delay characteristics matching those provided the current ARPANET user. Another issue is the qualification of current ARPANET users for AUTODIN II use: under the security requirements of AUTODIN II it seems that many ARPANET users will not qualify.

3. Make software and interface modifications to the IMPs and TIPs so that they can function as NFEs connected to AUTODIN II as a separate virtual subnetwork. This approach allows for the reuse of the ARPANET hardware, eliminates the need for separate ARPANET transmission facilities, and would enable DCA to transfer operational management to another suitable party.

> This strategy makes each IMP or TIP an AUTODIN II subscriber, replacing the IMP/modem interface with an IMP/PS interface, and requiring reprogramming of the IMPs and TIPs. A drawback with this approach is the cost of the reprogramming. Also, the previously mentioned security/qualification and performance issues pertain.

4. Transfer DoD users to AUTODIN II and transfer ARPANET transmission facility and operational management to receptive non-DoD Government agency (e.g., GSA). Gateway the ARPANET to AUTODIN II as necessary.

The separation of the current ARPANET users into two groups may have deeper implications than are readily apparent. The ARPANET provides capabilities for interaction between all its users, and in practice the communication between military users and nonmilitary users may be significant (for example, the communications between project sponsors and monitors in military organizations and project leaders and staff in university and corporate organizations). The thought that such a separation can be easily overcome via a gateway device is naive, as Section III shows. The provision for the small set of researchers working on network communication techniques may be jeopardized under this alternative.

5. Transfer DoD users to AUTODIN II and transfer transmission facility and operational management to a university consortium and possibly have an appropriate government agency (e.g., National Science Foundation) subsidize part of the operational costs. Gateway the ARPANET to AUTODIN II and Value Added Networks (VANs) as necessary.

Again the separation of the current ARPANET users into two groups and the provision for networking researchers are issues under this alternative. And again the idea of a gateway to solve intercommunication problems must be more thoroughly explored. The use of VANs is suggested, presumably for new users or those that don't qualify for AUTODIN II or the "university" ARPANET.

6. Transfer all desiring ARPANET users (hosts and terminals) who can satisfy the criteria of JCS MOP 165 to AUTODIN II. Transfer remaining users to a VAN (e.g., Telenet or Tymnet). Gateway AUTODIN II to VAN if necessary. Sponsors could dispose of the ARPANET hardware and DCA would no longer provide transmission for and operational management of the ARPANET.

This strategy comes closest to the main recommendation of this report, but again the separation of the current ARPANET users into two groups and the provision for networking researchers are issues under this alternative. And again the idea of a gateway to solve intercommunication problems must be more thoroughly explored. The VAN (or VANs) selected would have to provide service equivalent to the current ARPANET (as discussed in Sections II, III, and Appendix E). The hardware and software cost of interfacing the ARPANET hosts to the VANs is also of concern. The VANs are now adopting the CCITT X.25 interface, which is quite different from the ARPANET (BBN 1822) interface.

7. Transfer DoD users to AUTODIN II and arrange for a VAN to purchase the hardware assets of the ARPANET and to provide transmission facilities for and operational management of the ARPANET. Some government subsidization of the VAN may be justified. Gateway the ARPANET to AUTODIN II as necessary.

This strategy seems politically infeasible, and does not do away with any of the previously noted disadvantages.

Appendix G

Gateway Types

The following three types of gateways have been identified and used to characterize approaches to network interconnection. The main body of this report discusses gateways generally, while this appendix briefly comments on these three specific characterizations of gateways.

1. Hidden Gateway

This is a device that acts as a packet switch in one network (called net-A) and as a host or subscriber in another network (called net-B). A pair or collection of such devices can utilize net-B as all or part of the communication medium for net-A. A hidden gateway must take a raw packet of net-A and package it as a subscriber to subscriber message for transit through net-B to a complementary hidden gateway which unpacks the raw net-A packet.

This approach enables the shared use of communication media between two networks (e.g., long-haul high-bandwidth circuits), but does not enable users of one network to interact with users of the other. The device is called a hidden gateway because its existence is not apparent to users of either network. Net-B sees the device as just another subscriber, while Net-A sees the device as a packet switch and views net-B as a wire.

2. High Level Gateway

This is a device connected to two (or more) networks and provides a capability for interoperation across the network boundary. The extent to which the gateway participates in making the interoperation work is a measure of its level. A high level gateway transforms the protocols used in the source network to those used in the destination network. As discussed in Section III and Appendices A, B, and C, such transformations may be quite difficult.

3. Two-connected Host

In this approach a host (subscriber) is connected to both networks and equipped with the full range of network related software for both networks. The interaction between users on the two different networks is provided for by a set of application programs, one per functional area (e.g. Telnet, FTP). In effect the transformations of protocol differences are achieved by stripping off all packaging used in the first network, then repackaging using the conventions of the second network.

Acknowledgments

The authors would like to thank Dan Forsyth, who contributed to the thinking on the transition and the meaning of service equivalence and wrote Appendix E, and Keith Uncapher, who contributed helpful ideas throughout the project and helped to shape this report.